



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/596,769	06/23/2006	Roman Levenshteyn	P18511-US1	1296
27045 ERICSSON INC. 6300 LEGACY DRIVE M/S EVR 1-C-11 PLANO, TX 75024	7590 09/09/2009		<div>EXAMINER</div> <div>SURYAWANSI, SUDHISH K</div> <div>ART UNIT 2433</div> <div>PAPER NUMBER</div>	
			<div>MAIL DATE</div> <div>09/09/2009</div>	<div>DELIVERY MODE</div> <div>PAPER</div>

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

**Application No.**

10/596,769

**Applicant(s)**

LEVENSHTEYN ET AL.

**Examiner**

SUDHISH K. SURYAWANSI

**Art Unit**

2433

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 18 May 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/ICE)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

***Detailed Action***

1. This action is a final office action responsive to communication filed on: 18 May 2009 with acknowledgement of an original application filed on 23 June 2006.

2. **Claims 1-18** are currently pending. Claims 1, 7, and 13 are independent claims.

**Claims 1, 7, and 13-18** have been amended. Amendments to the claims are accepted.

**Claims 13-18** rejected under 35 U.S.C. 101 have been withdrawn based on Applicant's amendments.

**Claims 1-18** have been rejected.

***Response to Arguments***

3. Applicant's arguments filed 18 May 2009 have been fully considered however they are moot due to new grounds of rejection below.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 1-18 are rejected 35 U.S.C. 103(a)** as being unpatentable over *Jajodia S et al. U.S. Non-Literature Patent (NPL) "Flexible Support for Multiple Access Control Policies"*, Vol. 26, No. 2, June 2001, (hereinafter "*Jajodia*") in view of *Swift M M et al. U.S. Non-Literature (NPL) "Improving the Granularity of Access Control in Windows NT"*, May 3-4 2001, pages 87-96 (hereinafter "*Swift*").

**Rejection of Independent Claims**

**As to independent claim 1, Jajodia discloses "(Currently Amended) A method for taking a policy decision by a policy decision device"** (Paragraph titled "4.1 Architecture of the Authorization Framework" pages 228-229, fig. 7; reads on the limitations. Note that the authorization framework acts as a policy decision device which takes a policy decision based on the triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled "Definition 3.3 Authorization, lines 1-6.),

**"wherein the policy decision device has access to objects being relatable to each other by relations of one or more relation types the method comprising the steps of:"** (Paragraphs titled "Definitions 3.1, 3.2, Authorization Subject/Object Hierarchy", pages 222, and paragraph titled "Example 3.1", pages 224-226 until the end of the paragraph titled "Path overrides", fig. 5; reads on the limitations. Note that the policy decision device has access to objects based on the triple form as mentioned earlier – where objects refer to entities on which authorization can be specified..." could comprise person related data, computer devices, or application; the paragraph titled "3.1 Authorizations", page 222, lines 3-5),

**“receiving a request for the policy decision, the request specifying a first object of the objects and request information”** (fig. 7, triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled “Definition 3.3 Authorization, lines 1-6, and page 229, lines 10-11 – “...every request is seen as coming from either a user or a role...”; reads on the limitations. Note that the first object can be any object which refers to entities on which authorization can be specified with the request information in form of a as described on page 223, lines 7-14 (mail, faculty, + read) and (personal, faculty, -read), “This authorization states that the authorization subject faculty (second object), shown in fig. 4, can execute the action read (request) on the authorization object mail (first object) ...”),

**“said objects comprising entities that are controllable by one or more policies, said entities comprising person related data, computer devices, or applications”** (“objects refer to entities on which authorization can be specified...” could comprise person related data, computer devices, or application, page 222, lines 4-12 in the paragraph titled “3.1 Authorizations”, reads on the limitations. Note that any object that has an entity on which authorization can be specified is controllable by one or more policies.),

**“obtaining a policy matching to the request information and being applicable to a second object of the objects”** (Paragraph titled “Example 3.1”, page 224, lines 1-22; authorization for the subject (second object) G2 “(o, G2, -a)” in fig. 5; fig. 7, (authorization table), paragraph titled “chapter 4.1 Architecture of the Authorization

Framework”, page 228, lines 4-5; reads on the limitations. Note that the authorization framework obtains a policy (authorization propagation policy) matching -- based on the triple form and authorization table for the subject G2 (second object) -- to the request information (a) and being applicable to a second subject (G2 as the second object) of the objects (G1 to G5) in fig. 5, page 224.),

**“obtaining at least one propagation rule associated to the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier.),

**“the at least one propagation rule specifying at least one relation type of the one or more relation types”** (Paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier).

**The following are not explicitly taught in Jajodia: “verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of**

**the relation path are in accordance with at least one of the at least one specified relation type”** however Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled “4.1 Type-Specific Inheritance”, the entire paragraph, page 92; reads on the limitations; **and**

**“if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision”** Swift teaches the use of if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision in the paragraph titled “4.2 Static Inheritance”, the entire paragraph, page 92; reads on the limitations.

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention was made to include Swift in Jajodia, for verifying a path existence and applying the policy to the first object for taking the policy decision for the purpose of enabling fine-grained protection and allowing centralized management of object hierarchies by specifying more precisely how access control lists are inherited in Windows 2000 (Abstract).

**As to independent claim 7, Jajodia discloses “(Currently amended) A policy decision device for taking a policy decision, the policy decision device comprising:”** (Paragraph titled “4.1 Architecture of the Authorization Framework”

pages 228-229, fig. 7; reads on the limitations. Note that the authorization framework acts as a policy decision device which takes a policy decision based on the triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled "Definition 3.3 Authorization, lines 1-6.),

**"a receiving unit"** ("an inherited feature", fig. 7, pages 228-229, any of the components can act as a receiving unit, reads on the limitations) **and**

**"a processing unit"** ("an inherited feature", fig. 7, pages 228-229, any of the components can act as a processing unit, reads on the limitations), **wherein**

**"the processing unit is adapted to access accesses objects being relatable to each other by relations of one or more relation types"** (Paragraphs titled "Definitions 3.1, 3.2, Authorization Subject/Object Hierarchy", pages 222, and paragraph titled "Example 3.1", pages 224-226 until the end of the paragraph titled "Path overrides", fig. 5; reads on the limitations. Note that the processing unit which is an inherited feature within the policy decision device, has access to objects based on the triple form as mentioned earlier – where objects refer to entities on which authorization can be specified..." could comprise person related data, computer devices, or application; the paragraph titled "3.1 Authorizations", page 222, lines 3-5),

**"the receiving unit is adapted to receive-receives a request for the policy decision"** (fig. 7, triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled "Definition 3.3 Authorization, lines 1-6, and page 229, lines 10-11 –

"...every request is seen as coming from either a user or a role..."; reads on the limitations. Note that the first object can be any object which refers to entities on which authorization can be specified with the request information in form of a as described on page 223, lines 7-14 (mail, faculty, + read) and (personal, faculty, -read), "This authorization states that the authorization subject faculty (second object), shown in fig. 4, can execute the action read (request) on the authorization object mail (first object) ..."),

**"said objects comprising entities that are controllable by one or more policies, said entities comprising person related data, computer devices, or applications"**

("objects refer to entities on which authorization can be specified..." could comprise person related data, computer devices, or application, page 222, lines 4-12 in the paragraph titled "3.1 Authorizations", reads on the limitations. Note that any object that has an entity on which authorization can be specified is controllable by one or more policies.),

**"the processing unit is further adapted to obtain obtains a policy matching to the request information and being applicable to a second object of the objects"**

(Paragraph titled "Example 3.1", page 224, lines 1-22; authorization for the subject (second object) G2 "(o, G2, -a)" in fig. 5; fig. 7, (authorization table), paragraph titled "chapter 4.1 Architecture of the Authorization Framework", page 228, lines 4-5; reads on the limitations. Note that the processing unit is an inherited feature within the authorization framework which obtains a policy (authorization propagation policy) matching -- based on the triple form and authorization table for the subject G2 (second

object) -- to the request information (a) and being applicable to a second subject (G2 as the second object) of the objects (G1 to G5) in fig. 5, page 224.),

**“to obtain at least one propagation rule associated to the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides’, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier.),

**“the at least one propagation rule specifying at least one relation type of the one or more relation types”** (Paragraph titled ‘Path overrides’, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier).

**The following are not explicitly taught in Jajodia: “to verify if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, to verify if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type”** however Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more

of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled "4.1 Type-Specific Inheritance", the entire paragraph, page 92; reads on the limitations; **and**

**"if said relation path exists and if said one or more relations of the relation path are in accordance, to apply the policy to the first object for taking the policy decision"** Swift teaches the use of if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision in the paragraph titled "4.2 Static Inheritance", the entire paragraph, page 92; reads on the limitations.

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention was made to include Swift in Jajodia, for verifying a path existence and applying the policy to the first object for taking the policy decision for the purpose of enabling fine-grained protection and allowing centralized management of object hierarchies by specifying more precisely how access control lists are inherited in Windows 2000 (Abstract).

**As to independent claim 13, Jajodia discloses "(Currently amended) A computer ~~program~~ into-readable medium** (page 257, lines 2-8 reads on the limitations of a computer readable medium) **of a policy decision device, the computer ~~program~~-readable medium ~~comprising code adapted~~ having stored thereon a plurality of instructions, the plurality of instructions including instructions which, when executed by a processor, cause the processor to**

**perform the steps of a method for taking a policy decision, comprising:"**

(Paragraph titled "4.1 Architecture of the Authorization Framework" pages 228-229, fig. 7; reads on the limitations. Note that the authorization framework acts as a policy decision device which takes a policy decision based on the triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled "Definition 3.3 Authorization, lines 1-6.)

**"accessing to objects being relatable to each other by relations of one or more relation types"** (Paragraphs titled "Definitions 3.1, 3.2, Authorization Subject/Object Hierarchy", pages 222, and paragraph titled "Example 3.1", pages 224-226 until the end of the paragraph titled "Path overrides", fig. 5; reads on the limitations. Note that the policy decision device has access to objects based on the triple form as mentioned earlier – where objects refer to entities on which authorization can be specified..." could comprise person related data, computer devices, or application; the paragraph titled "3.1 Authorizations", page 222, lines 3-5),

**"the request specifying a first object of the objects and request information"**(fig. 7, triple form (o, s, (sign) a) where o is a second object, s is subject as a first object, and a is a request and the sign (+ or -) is an action, page 223 in the paragraph titled "Definition 3.3 Authorization, lines 1-6, and page 229, lines 10-11 – "...every request is seen as coming from either a user or a role..."; reads on the limitations. Note that the first object can be any object which refers to entities on which authorization can be specified with the request information in form of a as described on

page 223, lines 7-14 (mail, faculty, + read) and (personal, faculty, -read), "This authorization states that the authorization subject faculty (second object), shown in fig. 4, can execute the action read (request) on the authorization object mail (first object) ..."),

**"said objects comprising entities that are controllable by one or more policies, said entities comprising person related data, computer devices, or applications"** ("objects refer to entities on which authorization can be specified..." could comprise person related data, computer devices, or application, page 222, lines 4-12 in the paragraph titled "3.1 Authorizations", reads on the limitations. Note that any object that has an entity on which authorization can be specified is controllable by one or more policies.),

**"obtaining a policy matching to the request information and being applicable to a second object of the objects"** (Paragraph titled "Example 3.1", page 224, lines 1-22; authorization for the subject (second object) G2 "(o, G2, -a)" in fig. 5; fig. 7, (authorization table), paragraph titled "chapter 4.1 Architecture of the Authorization Framework", page 228, lines 4-5; reads on the limitations. Note that the authorization framework obtains a policy (authorization propagation policy) matching -- based on the triple form and authorization table for the subject G2 (second object) -- to the request information (a) and being applicable to a second subject (G2 as the second object) of the objects (G1 to G5) in fig. 5, page 224.),

**"obtaining at least one propagation rule associated to the policy"** (Paragraph titled "Example 3.1", page 224, lines 10-11 "We are now ready to describe various

different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier.),

**“the at least one propagation rule specifying at least one relation type of the one or more relation types”** (Paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier).

**The following are not explicitly taught in Jajodia: “verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type”** however Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled “4.1 Type-Specific Inheritance”, the entire paragraph, page 92; reads on the limitations; **and**

**“if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision”** Swift teaches the use of if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision in the paragraph titled “4.2 Static Inheritance”, the entire paragraph, page 92; reads on the limitations.

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention was made to include Swift in Jajodia, for verifying a path existence and applying the policy to the first object for taking the policy decision for the purpose of enabling fine-grained protection and allowing centralized management of object hierarchies by specifying more precisely how access control lists are inherited in Windows 2000 (Abstract).

**Rejection of Dependent Claims**

**As to dependent claim 2, Jajodia discloses “(Previously Presented) The method according to claim 1, wherein the at least one propagation rule specifies at least one direction”** (Paragraph titled ‘Path overrides’, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier), **and**

**“the method further comprises the step of verifying if the one or more relations of the relation path are in accordance with the at least one specified direction”** Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled “4.1 Type-Specific Inheritance”, the entire paragraph, page 92; reads on the limitations.

**As to dependent claim 3, Jajodia discloses “(Previously Presented) The method according to claim 1, wherein the at least one propagation rule specifies at least one condition** (Paragraph titled ‘Path overrides’, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier) **that is verified for at least one of the objects of the relation path”** (Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled “4.1 Type-Specific Inheritance”, the entire paragraph, page 92; reads on the limitations).

**As to dependent claim 4, Jajodia discloses “(Previously Presented) The method according to claim 1, wherein the existence of the relation path is**

**considered for obtaining the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 5, Jajodia discloses “(Previously Presented) The method according to claim 1, wherein the at least one propagation rule is obtained from at least one propagation rule database on the basis of at least one reference identifier associated to the at least one propagation rule and the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 6, Jajodia discloses “(Previously Presented) The method according to claim 1, wherein at least one further policy component of the policy is obtained from at least one policy component database based on at least one reference identifier associated to the at least one further policy component and the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11

"We are now ready to describe various different authorization propagation policies...", paragraph titled 'Path overrides', page 226, lines 6-7, "Authorizations of a node are propagated to its subnodes if not overridden..."; reads on the limitations. Note that a certain action (+a or -a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 8, Jajodia discloses "(Previously Presented) The policy decision device according to claim 7, wherein the at least one propagation rule specifies at least one direction** (Paragraph titled 'Path overrides', page 226, lines 6-7, "Authorizations of a node are propagated to its subnodes if not overridden..."; reads on the limitations. Note that a certain action (+a or -a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier) **and the processing unit is adapted to verify if the one or more relations of the relation path are in accordance with the at least one specified direction"** (Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled "4.1 Type-Specific Inheritance", the entire paragraph, page 92; reads on the limitations).

**As to dependent claim 9, Jajodia discloses "(Previously Presented) The policy decision device according to claim 7, wherein the at least one propagation rule**

**specifies at least one condition** (Paragraph titled "Path overrides", page 226, lines 6-7, "Authorizations of a node are propagated to its subnodes if not overridden..."; reads on the limitations. Note that a certain action (+a or -a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier) **and the processing unit is adapted to verify, for at least one of the objects of the relation path, if said at least one object is in accordance with the at least one condition"** (Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled "4.1 Type-Specific Inheritance", the entire paragraph, page 92; reads on the limitations).

**As to dependent claim 10, Jajodia discloses "(Previously Presented) The policy decision device according to claim 7, wherein the processing unit is adapted to consider the existence of the relation path for the obtaining of the policy"** Swift teaches the use of if said relation path exists and if said one or more relations of the relation path are in accordance, applying the policy to the first object for taking the policy decision in the paragraph titled "4.2 Static Inheritance", the entire paragraph, page 92; reads on the limitations.

**As to dependent claim 11, Jajodia discloses "(Previously Presented) The policy decision device according to claim 7, wherein the processing unit is adapted to obtain the at least one propagation rule from at least one propagation**

**rule database on the basis of at least one reference identifier associated to the at least one propagation rule and the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 12, Jajodia discloses “(Previously Presented) The policy decision device according to claim 7, wherein the processing unit is adapted to obtain at least one further policy component of the policy from the at least one policy component database based on at least one reference identifier associated to the at least one further policy component and the policy”**

(Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 14, Jajodia discloses “(Currently amended) The computer program-readable medium (page 257, lines 2-8 reads on the limitations of a computer readable medium) according to claim 13, wherein the at**

**least one propagation rule specifies at least one direction"** (Paragraph titled 'Path overrides', page 226, lines 6-7, "Authorizations of a node are propagated to its subnodes if not overridden..."; reads on the limitations. Note that a certain action (+a or -a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule specifying at least one relation type of the one or more relation types as described earlier) and

**"the computer program comprises code adapted to verify if the one or more relations of the relation path are in accordance with the at least one specified direction"** Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance with at least one of the at least one specified relation type in the paragraph titled "4.1 Type-Specific Inheritance", the entire paragraph, page 92; reads on the limitations.

**As to dependent claim 15, Jajodia discloses "(Currently amended)**  
**The computer ~~program~~ readable medium** (page 257, lines 2-8 reads on the limitations of a computer readable medium) **according to claim 13, wherein the at least one propagation rule specifies at least one condition and the computer program comprises code adapted to verify for at least one of the objects of the relation path if said at least one object is in accordance with the at least one condition"** Swift teaches the use of verifying if a relation path exists, the relation path linking the first object and the second object and consisting of one or more of the relations, verifying if the one or more relations of the relation path are in accordance

with at least one of the at least one specified relation type in the paragraph titled "4.1 Type-Specific Inheritance", the entire paragraph, page 92; reads on the limitations.

**As to dependent claim 16, Jajodia discloses "(Currently amended)**

**The computer ~~program~~-readable medium** (page 257, lines 2-8 reads on the limitations of a computer readable medium) **according to claim 13, wherein the computer program comprises code adapted to consider the existence of the relation path for obtaining the policy".**

**As to dependent claim 17, Jajodia discloses "(Currently amended)**

**The computer ~~program~~-readable medium** (page 257, lines 2-8 reads on the limitations of a computer readable medium) **according to claim 13, wherein the computer program comprises code adapted to obtain the at least one propagation rule from at least one propagation rule database on the base of at least one reference identifier associated to the at least one propagation rule and the policy"** (Paragraph titled "Example 3.1", page 224, lines 10-11 "We are now ready to describe various different authorization propagation policies...", paragraph titled 'Path overrides', page 226, lines 6-7, "Authorizations of a node are propagated to its subnodes if not overridden..."; reads on the limitations. Note that a certain action (+a or -a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier).

**As to dependent claim 18, Jajodia discloses "(Currently amended)**

**The computer ~~program~~-readable medium** (page 257, lines 2-8 reads on the limitations of a computer readable medium) **according to claim 13, wherein the**

**computer program comprises code adapted to obtain at least one further policy component of the policy from the at least one policy component database based on at least one reference identifier associated to the at least one further policy component and the policy”** (Paragraph titled “Example 3.1”, page 224, lines 10-11 “We are now ready to describe various different authorization propagation policies...”, paragraph titled ‘Path overrides”, page 226, lines 6-7, “Authorizations of a node are propagated to its subnodes if not overridden...”; reads on the limitations. Note that a certain action (+a or –a) is taken based on the received request within a triple form (o, s, (sign) a) which also contains propagation rule associated to the policy as described earlier.).

### ***Conclusion***

Applicant’s amendment necessitated the new ground(s) of rejection presented in this Office Action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. It is noted, PATENTS ARE RELEVANT AS PRIOR ART FOR ALL THEY CONTAIN "The use of patents as references is not limited to what the patentees describe as their own inventions or to the problems with which they are concerned. They are part of the literature of the art, relevant for all they contain." In re Heck, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006, 1009, 158 USPQ 275, 277 (CCPA 1968)). A reference may be relied upon for all that it would have reasonably suggested to one having ordinary skill the art, including nonpreferred embodiments (see MPEP 2123).

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sudhish K Suryawansi whose telephone number is (571) 270-7461. The examiner can normally be reached Monday through Friday between 9 am to 5 pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Moazzami can be reached on (571) 272-4195. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should

Art Unit: 2433

you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/SUDHISH SURYAWANSI/  
Patent Examiner, Art Unit 2433

September 03, 2009

/Nasser G Moazzami/

Supervisory Patent Examiner, Art Unit 2436